

日本語印刷文書における文字切り出し†

—新聞自動読み取りへの応用—

豊田 順一††* 野口 要治††** 西村 康††

本論文は、日本語印刷文書における文字切り出し手法について述べたものである。日本語印刷文書では、1行あたりの文字数を1字増やしたり、減らしたりすることがしばしば行われる。これは、追い込み、追い出し操作と呼ばれるものであり、文字ピッチが大きく乱れる原因となっている。したがって、黒画素の一次元射影を調べ、文字ピッチが一定であるという先験的知識のみに依存して文字切り出しを行う従来の手法では、文字切り出しの成功率は低いと思われる。著者らの手法は、まず文字ピッチの乱れを検出し、さらにそれが追い込み、あるいは追い出し操作によるものか否かを判定する。もしそうであれば、それらの操作が行われた場合の文字配置に関する知識を利用して、文字切り出しを行う。このような知識の利用とともに、従来の画素単位の処理に加えて、処理の早い段階から、1文字に相当する画素の集合を処理の単位とするために、新たに8連結ブロックと呼ぶ黒画素の集合を定義し、8連結ブロックの大小とそれらの相対的位置関係を利用して文字切り出しを行う点に大きな特徴がある。著者らは、新聞などの自動読み取りシステムについて研究中であり、この手法は、そのシステムの一部分として新聞を対象に開発したが、新聞に限らず他の日本語印刷文書にも十分適用できると考えられる。

1. ま え が き

新聞によって運ばれる情報の量はきわめて多く、その有効利用に対する要求は非常に大きい。そして、このような事態は今後も変化しないであろう。新聞を計算機で処理するためには、いったん新聞紙面を画像として入力し、記事を機械的に読み取る機構を実現することが必要である。このような技術は文書画像処理と称せられ、近年盛んに研究されるようになってきた。

新聞紙面には、図、写真あるいは文字等が同時に存在しているので、個々の文字の認識に先立ち、各領域の同定や文章のつながりの追跡を行う必要がある。すなわち、新聞のもつ書式構造の認識にまで立ち入ることが必要となる。これについては、数多くの研究発表が行われており、新聞紙面の書式構造に関する知識を利用して各記事を切り抜く手法が報告されている^{1)~4)}。書式構造の認識と個々の文字の抽出ができればすでに提案されたパターン認識の手法⁵⁾を利用して、記事を構成する文字部分を読み取ることが可能である。

著者らは新聞等の自動読み取り器の開発を最終目標として、研究を進めてきた。本論文では、このシステ

ムのうち、文字認識の前段階に位置する文字切り出し手法について述べる⁶⁾。新聞記事等の日本語印刷文書では、文字ピッチの不整、文字同士の接触、あるいは紙面の汚れが完全な文字切り出しを困難なものにしている。従来の手法^{7)~9)}が共通に利用しているのは文字を構成する黒画素の一次元射影と文字ピッチが一定であるとの仮定である。一方、日本語印刷文書では版を組む際に、行の文字数を変える“追い込み”や“追い出し”と呼ばれる操作が行われる場合があり、このような操作が行われた行では、文字ピッチが一定であるとの仮定に基づく文字切り出しは成功しないことが多い。また追い込み操作の対象は“o”等の文構成要素であり、これらは文中で重要な役割を果たす。

本論文で提案する文字切り出し手法は、追い込みあるいは追い出し操作が行われた場合の文字配置に関する知識を利用して、それらの操作が行われた行での文字切り出しを有効にした。それとともに、従来の画素単位の処理に加えて、新たに“8連結ブロック”と呼ぶ黒画素の集合を定義し、8連結ブロックの大小とそれらの相対的位置を利用して処理を行っている。著者らの手法は、一般の日本語印刷文書に関する知識を利用しているため、新聞に限らず他の日本語印刷文書にも適用できると考えられる。

2. 組版規則と追い込み、追い出し操作

日本語印刷文書の版をつくる際の組版規則について簡単に説明し、次にこの組版規則に従って版を組む過

† Segmentation of Machine Printed Japanese Text—Its Application to a Newspaper Reader by JUNICHI TOYODA, YOHJI NOGUCHI and YASUSHI NISHIMURA (Department of Information and Computer Science, Faculty of Engineering Science, Osaka University)

†† 大阪大学基礎工学部情報工学科

* 現在 大阪大学産業科学研究所

** 現在 シャープ(株)



(a) 追込み操作の行われた行 (b) 追出し操作の行われた行

図1 追込みおよび追出し操作が行われた行の例
Fig. 1 Example of lines whose character numbers are changed.

で行われる追込み,あるいは追出しと呼ばれる操作を説明する。

1行に一定数の文字を埋めようとすれば,以下の組版規則を考慮する必要がある¹⁰⁾。

(1) 「, », 「, 「, 「など開きカッコを行末においてはならない。

(2) 「, 「, 「, 「など閉じカッコや,句読点,促音あるいは拗音などを行頭においてはならない。

(3) 時間,年号あるいは距離などを表す数字列が2行にまたがってはならない。

このような規則に従って版を組むために行われるのが追込みあるいは追出し操作である。

追込みは,1行中に「,「,「,「等の縦の長さが短い文字があったときに,その上下の文字との間隔をせばめて半角にし,1行中に埋め込む文字を一つ増やす操作である。追込み操作の行われた行では,このように上下の間隔をせばめられた文字が2文字存在する。図1(a)に追込み操作の行われた行の例を示す。

追出しは,逆に文字間隔を広げて1行中に埋め込む文字を一つ減らす操作である。図1(b)に追出し操作が行われた行の例を示す。

このような追込みおよび追出し操作は,日本語印刷文書において文字ピッチが大幅に乱れる原因となっている。

3. 8連結ブロック

文字切り出しに際しては,処理時間を短縮するため入力2値画像の参照は必要な時にのみ行うことにし,以下で説明する8連結ブロックを対象として処理を行う。

入力2値画像の各画素は,画像の左上隅を原点($x=0, y=0$)とする座標(i, j)をもつとし,この画素

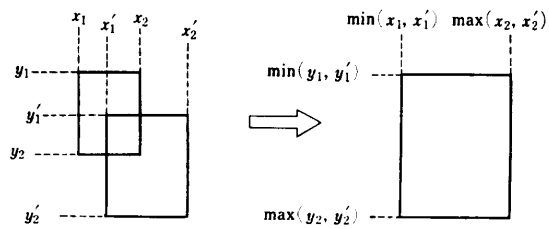


図2 8連結ブロックの生成操作
Fig. 2 Process of yielding 8-connected block.

を(i, j)で表す。

2値画像上の黒画素連結領域の一つを $d = \{(i, j)\}$ としたとき,黒画素8連結領域を囲む最小矩形を以下のように定義する。 x_1, x_2, y_1 および y_2 をそれぞれ

$$x_1 = \min \{i \mid (i, j) \in d\}$$

$$x_2 = \max \{i \mid (i, j) \in d\}$$

$$y_1 = \min \{j \mid (i, j) \in d\}$$

$$y_2 = \max \{j \mid (i, j) \in d\}$$

とする。 $(x_1, y_1), (x_1, y_2), (x_2, y_1)$ および (x_2, y_2) を四隅の座標とする矩形を4字組 (x_1, x_2, y_1, y_2) で表し,これを8連結領域を囲む最小矩形とする。

次の処理によって8連結ブロックを得る。まず,入力2値画像から8連結領域を囲む最小矩形をすべて求め,それらの最小矩形の集合を D とする。 D に属する二つの要素 (x_1, x_2, y_1, y_2) および (x_1', x_2', y_1', y_2') について

$$(x_1 \leq x_2') \wedge (x_2 \geq x_1') \wedge (y_1 \leq y_2') \wedge (y_2 \geq y_1')$$

なる関係が成立する場合,つまり二つの矩形が互いに交わるとき,図2のようにこの二つの矩形を統合した矩形,

$$(\min(x_1, x_1'), \max(x_2, x_2'), \min(y_1, y_1'), \max(y_2, y_2'))$$

を D に加え,元の二つの矩形を D から取りのぞく。この操作を D のいずれの二つの矩形も互いに交わらなくなるまで繰り返す。その結果得られた矩形が8連結ブロックであり,その集合 D を8連結ブロックの集合と名づける。

個々の8連結ブロックは,4字組 (x_1, x_2, y_1, y_2) で表すことにする。また2値画像上の任意の黒画素は8連結ブロックのいずれかによって囲まれ,かつ1個の8連結ブロックに囲まれる。

4. 文字切り出しアルゴリズム

4.1 文字切り出し処理の流れ

文字切り出し処理は,新聞紙面から個々の記事が切

り抜かれた後、記事中の文字を1個ずつ抽出するものとして位置づけることができる。

文字切り出し処理のため必要な入力、新聞紙面から得た2値画像、この2値画像から得る8連結ブロックの集合、および文字切り出し対象となる本文または抄録文領域である。新聞紙面上で本文の文字は15の段に分けて配置されているが、ここで示す本文領域とは一つの記事本文のうち1段に置かれた部分である。

文字切り出しプログラムは、これらの入力に対して (i) 雑音除去、(ii) 使用されている活字サイズの決定、(iii) 行の抽出、(iv) 行単位の文字切り出しの順に処理を行う。いずれの段階でも、8連結ブロックの集合を対象に処理を行い、(iv) の行単位の文字切り出しでは必要な場合のみ2値画像を参照する。

処理の結果、出力として個々の文字を囲む最小矩形のリストを出力する。

4.2 雑音除去

8連結ブロックの集合 D 中の要素 (x_1, x_2, y_1, y_2) について、

$$(x_1 = x_2) \wedge (y_1 = y_2)$$

なる関係が成立するならば、この8連結ブロックは、2値画像上の孤立点から生じたものである。

したがって、文字切り出し処理を行う際に雑音となるため、 D の要素のうち上記の関係を満たすものを D から取りのぞく。以下の処理で用いる8連結ブロックの集合 D は、この処理で雑音除去が行われたものである。

4.3 活字サイズの決定

新聞記事の本文および抄録文で用いられる活字は、「倍」と呼ばれる $2.24\text{mm} \times 2.80\text{mm}$ の大きさのものが主流である。全体の約30%の新聞では、この活字より10%程度大きな活字が用いられており、また同一の紙面で数種類の活字が使い分けられる場合もある。

これに対処するため、本文あるいは抄録文領域に存在する8連結ブロックには、一つの文字を囲む最小矩形となっているものが多いことを利用して、使われている活字の大きさを決定する。すなわち、あらかじめ「倍」の大きさの活字、あるいはそれより10%程度大きな活字の標準的な大きさを与えておき、次に本文あるいは抄録文領域内に存在する8連結ブロックの大きさを調べる。あらかじめ与えた各活字の標準的な大きさと等しい大きさをもつ8連結ブロック数をそれぞれ累算し、この累算値が最大となったものがその領域

で用いられている活字サイズと決定する。

ひとたび使用されている活字サイズが決定されると文字切り出しプログラム中で用いるいくつかのパラメータ、すなわち標準的な文字の横幅の最小値 l_x 、標準的な文字長さの最小値 l_{y_1} 、および文字長さの最大値 l_{y_2} などを、使用されている活字に対応する値にそれぞれ設定する。たとえば「倍」の大きさの活字であれば、 $l_x = 20$ 、 $l_{y_1} = 15$ 、 $l_{y_2} = 17$ (単位 $1/8\text{mm}$) とする。

4.4 行の抽出

新聞記事の本文および抄録文領域に属する文字が縦方向に連続的に配置されていることを利用して、これらの領域を構成する各行を抽出する。

与えられた本文あるいは抄録文領域に属する8連結ブロックのうち、互いの縦方向距離が $2 \cdot l_{y_1}$ 以下のものを3章で述べた方法で統合し、得られた8連結ブロックを元の二つの8連結ブロックに替える。この操作を繰り返した結果得られる8連結ブロックのうち、

$$((x_2 - x_1) \geq l_x) \wedge ((y_2 - y_1) \geq l_{y_1})$$

なる関係を満たすものを行とする。

4.5 行単位の文字切り出し

4.5.1 処理のあらまし

この節では、一つの行を対象とした文字切り出し処理を述べる。この行単位の処理を一つの本文あるいは抄録文領域のすべての行にほどこすことにより、文字切り出しが終了する。

まず、ある行に属する8連結ブロックで横方向に隣り合うものを統合する補助的処理を行う。次に文字ピ

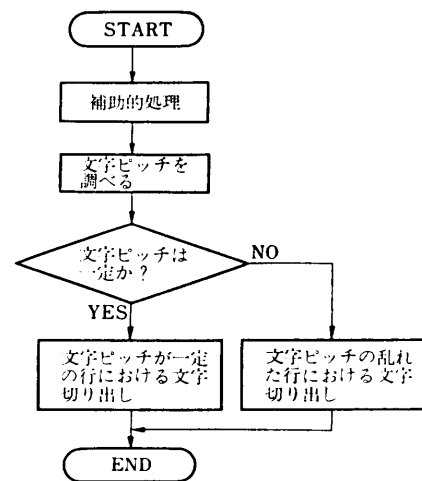


図3 各行での文字切り出し処理の流れ
Fig. 3 Segmentation algorithm for one line.

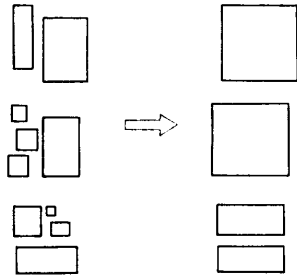


図4 横方向に隣り合う8連結ブロックの統合処理
Fig. 4 Merging process of 8-connected blocks.

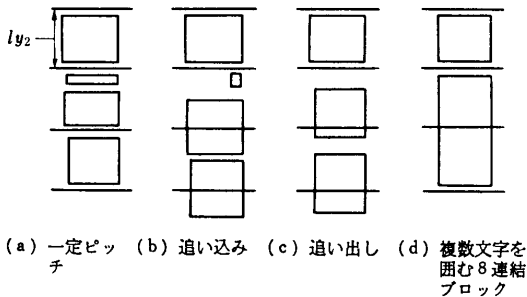


図5 文字ピッチの乱れ検出
Fig. 5 Detecting pitch variation.

ッチの乱れ検出を行い、文字ピッチの乱れがなければ文字ピッチに基づいて各文字を切り出す。文字ピッチの乱れが検出された場合には、追いつみあるいは追いつき操作が行われたか否かを判断し、組版規則に基づく文字ならびに関する知識を用いて各文字を切り出す。図3に行単位の文字切り出し処理の全体を示す。

4.5.2 文字切り出しに先立つ補助的処理

“行”や“信”など横方向に分離した文字や、線切れを起こした文字は、一つの8連結ブロックとならず横方向に隣り合う複数個の8連結ブロックとなる。このような8連結ブロックを統合する。この処理で1文字に対応する8連結ブロック数を増やすことにより、文字切り出しが容易に行え、また処理時間が減少する。また、この処理によって文字切り出しに必要な情報が損なわれることはない。図4にこの処理の例を示す。この処理の結果、同一行に存在する8連結ブロックで互いに横方向に隣り合うものは存在しなくなる。

4.5.3 文字ピッチの乱れ検出

ある行において、文字ピッチが一定であるか否かを調べるために4.3節で得たパラメータ l_{y_2} 間隔で仮想的な横線を引き、その横線と各8連結ブロックとの重なり具合を調べることにする。文字ピッチが一定の行においては、横線と大きく交わる8連結ブロックは存在しない。ある横線がいずれかの8連結ブロックと交

わり、しかもその8連結ブロックの上辺および下辺から横線までの距離がそれぞれ $(1/3) \cdot l_{y_2}$ を超えるようなことが起これば、その行での文字ピッチは乱れていると判断する。図5にこの概念図を示す。

4.5.4 文字ピッチが一定の行での文字切り出し

文字ピッチが一定と判断した行では、次のような手順に従って上から順に文字を切り出していく。

まず、行の始まりに位置する文字を切り出すため、

4.5.3項で引いた仮想的な横線にはさまれた領域のうち、最も上に存在する領域に属する8連結ブロックを統合して、行の先頭の文字とする。次に、以下に示す上方からの切り出し処理を行う。いま得た文字の下辺座標 y_2 といま用いた領域の下辺の横線座標 Y を比べ

$$y_2 > Y - (1/5) \cdot l_{y_2}$$

なる関係が存在すれば、次の文字を切り出すための領域を y_2 と $y_2 + l_{y_2}$ の間に設定し、そうでなければ Y と $Y + l_{y_2}$ の間に設定する。こうして得た領域内の8連結ブロックを統合し、それを一つの文字として切り出す。上記の、仮想的な横線にはさまれた領域の設定、およびその領域内に含まれる8連結ブロックの統合による1文字切り出し過程をその行に存在するすべての8連結ブロックが尽きるまで繰り返す。

ここで述べた処理は、対象とする行にわずかな文字ピッチの変動があっても、正しく文字を切り出すことができる。

4.5.5 文字ピッチの乱れた行での文字切り出し

文字ピッチの乱れた行では、まず1文字分の大きさをもつすべての8連結ブロックをさがし出し、それらを手掛りとして文字切り出しを始めることにする。一つの8連結ブロック (x_1, x_2, y_1, y_2) について、

$$((x_2 - x_1) > l_x) \wedge (l_{y_1} < (y_2 - y_1) < l_{y_2})$$

が成立すれば、 (x_1, x_2, y_1, y_2) は1文字分の大きさをもつ8連結ブロックであるとする。対象とする行に存在するこのような8連結ブロックをそれぞれ1文字として切り出す。

この処理を終えた後に、その行の先頭の文字が切り出せていなければ、それを切り出す。行頭の文字は、段落の先頭の行では本文または抄録文領域の上端から1文字分下がった位置に置かれ、そうでなければこれらの領域の上端に接して置かれることを利用し、これらの領域の上端座標 Y と $Y + l_{y_2}$ (あるいは $Y + l_{y_2}$ と $Y + 2 \cdot l_{y_2}$) との間に存在するすべての8連結ブロックを統合してできる8連結ブロックを1文字とする。

この段階で、対象となる行のすべての文字がすでに切り出せたときには、この行における処理を完了す

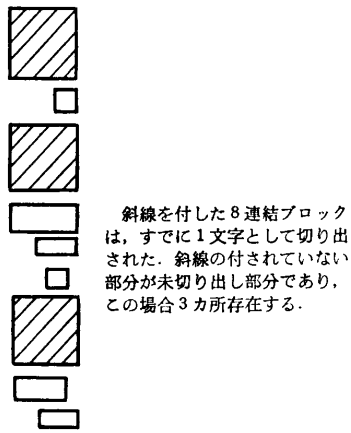


図 6 先頭文字と適当な大きさの8連結ブロックを切り出した段階での行の様子
Fig. 6 Existence of segmented and unsegmented part.

る。そうでなければ、対象とする行には、すでに文字であるとして切り出した8連結ブロックの間はいくつかの未切り出し部分が存在する。未切り出し部分とは、ある処理を行った段階で、各文字を切り出せなかった部分のことであり、一般に対象とする行に複数個存在する。また、未切り出し部分は行の途中にあるか、あるいは行尾に存在する。この段階まで処理の進んだ行の様子を図6に示す。

このように未切り出し部分が残った行からさらに個々の文字を切り出す処理を行うが、この部分は処理の流れが複雑なため、処理全体の流れをまず図7に示し図中で番号を付けた処理について説明する。

(1) 追い込み、追い出し操作の有無を調べる処理
ここでは、文字配置に関する知識を利用し、すでに切り出した文字間の位置関係調べることにより、追い込みあるいは追い出し操作が行われた行であるか否かを判断する。

以下に示す条件を満たす文字の並びが存在するとき

その部分で追い込み操作が行われたと判断する。条件とは、一つの文字が切り出されており、その文字が標準的な1文字分の大きさ(文字幅 l_x 以上、文字長さ l_{v1} 以上 l_{v2} 以下)をもち、その文字の直下に8連結ブロックが一つ存在し、それにひき続いて一つの文字が切り出されており、その大きさが標準的な1文字分と見なせ、さらに上側の文字の下辺と下側の文字の上辺の距離が $(3/4) \cdot l_{v1}$ より小さいことである。このとき、中央の8連結ブロックを1文字であるとし、この文字に対して追い込み操作が行われたと判断する。一般に追い込み操作が行われた行では、このような文字が二つ存在する。

また引き続き2文字が切り出されており、2文字とも標準的な1文字分の大きさをもち、さらに上側の文字の下辺と下側の文字の上辺の距離が $(1/3) \cdot l_{v1}$ より大きい場合、追い出し操作が行われたと判断する。

(2) 追い出し操作が行われた行、あるいは追い込み操作が行われ、その対象となった2文字

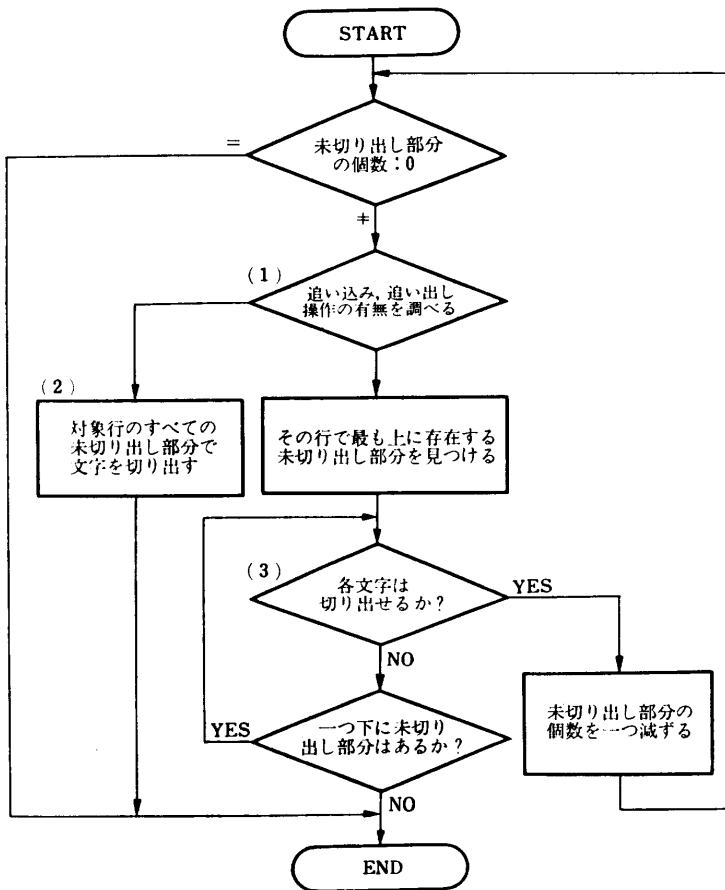


図 7 未切り出し部分での文字切り出し処理の流れ
Fig. 7 Flow of processing unsegmented parts of text.

を発見した場合の処理

この場合には、4.5.4項と同様の上方からの切り出し処理を行い、すべての文字を切り出す。

個々の未切り出し部分の直上の文字の下辺座標 y を起点として、文字切り出し領域の設定、および1文字切り出しの過程をその未切り出し部分のすべての8連結ブロックが尽きるまで繰り返す。ここでの処理では y と $y+l_v$ とにはさまれた領域に一部分でも存在する8連結ブロックを統合し、得られた8連結ブロックの長さが l_v を超えるときには、(i)この8連結ブロックの上辺座標 y_1 と y との間に

$$y_1 - y > (1/5) \cdot l_v,$$

なる関係が成立すれば、 y_1+l_v を中心とする5画素の範囲で、(ii)そうでなければ、 $y+l_v$ を中心とする5画素の範囲で、入力2値画像を参照して対象とする行を横方向に走査して黒画素の周辺分布を求め、黒画素の累積値が最小となる点で8連結ブロックを上下に分割して上側のものを1文字として切り出す。

(3) 一つの未切り出し部分で個々の文字を切り出せるか否か判断する処理

処理(1)で、追い込み操作が行われ、その対象となった1文字を発見した場合、および追い込み、追い出し操作を発見できなかった場合には、個々の未切り出し部分で切り出しを行えるか否か調べる。

未切り出し部分の長さが l_v 以下であれば、その部分に存在する8連結ブロックを統合した上で、それを1文字として切り出す。

未切り出し部分の長さが l_v 以上であれば(2)で述べた上方からの文字切り出しと、それと相補的に下方から文字切り出しを行った結果を比較する。両方の結果が一致すれば、切り出し結果に基づき、各文字を切り出す。対象とした未切り出し部分で、文字ピッチが一定の場合、および文字間隔が開く方向に文字ピッチの変動がある場合には、その長さが l_v 以上であっても個々の文字を切り出せる。

処理(3)で、対象とする未切り出し部分で個々の文字を切り出せた場合には、未切り出し部分の個数を一つ減じ、対象行にまだ未切り出し部分が存在する場合には、処理(1)を行う。対象行に未切り出し部分が存在しない場合には、この行での処理を終える。

処理(3)で、対象とする未切り出し部分で個々の文字を切り出せないと判断した場合には、図7に示したように、いま切り出しを試みた部分の直下にある未切り出し部分に対して、処理(3)を行う。もし、直下に

未切り出し部分がなければ、すなわちいま切り出しを試みた部分が対象とする行で最も下方に位置する未切り出し部分であったなら、その行のすべての未切り出し部分の位置とそれぞれの未切り出し部分に属する8連結ブロックを記憶して、この行での処理を終える。

5. 実験結果と検討

新聞紙面をファクシミリ(8画素/mm)により入力し、2値画像化された記事について文字切り出しを試みた。三大紙の記事本文領域に属する合計8,967文字について実験を行ったところ、活字サイズ決定結果の正答率100%、行抽出結果の正答率97.7%、さらに文字を切り出せた割合95.9%、文字を切り出した箇所における正答率99.0%を得た。また、同じデータに対して未切り出し部分の各文字を強制的に切り出したところ正答率98.7%を得た。比較のため同じデータに対して文献6)で示した手法を用いて文字切り出しを行ったところ正答率98.0%を得た。なお、このデータにおいて追い込みあるいは追い出し操作によって文字ピッチの乱れた文字は、全体の2.7%であったが、これらの文字に対する文献6)の切り出し手法の正答率は82.1%、本論文の手法では87.4%となった。この結果から著者らの手法は、従来の手法に比べて有効であるといえよう。図8に本論文で示した手法による処理結果の一例を、切り出した各文字を矩形で囲んで示す。

文字を切り出せない原因は、大きく分けて二つ考えられる。すなわち、(1)印刷インクの飛沫等による雑音、文字のかすれ、線切れおよびつぶれなどのために、(a)追い込みあるいは追い出し操作が行われたにもかかわらず、それらが発見できなかったこと、あるいは(b)複数個の文字を囲んでいると考えられる8連結ブロックについての上方および下方からの切り出し結果が一致しないため、文字切り出しアルゴリズムが働かなかったこと等である。また、(ii)追い込みが

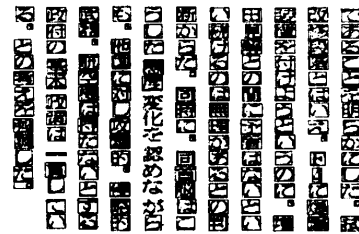


図8 実験結果の一例

Fig. 8 Example of segmented text.

行われたことを発見し、その対象となった対をなす文字の片方を見いだした。もう一方の文字を含む部分の文字並びの様子からは、どれがその文字であるかを同定できなかったこと等である。

誤った文字切り出しが行われる原因は、複数個の文字を囲む8連結ブロックを分割する際に黒画素の周辺分布に依存していることである。文字の種類によっては、文字が接触する点で周辺分布が最小になるとは限らず、誤りの原因となっている。これについては現在検討中である。

6. むすび

新聞記事の本文および抄録文領域を対象とする文字切り出しについて述べた。この手法では追い込みや追い出し操作の行われた行での文字配置に関する知識を用いて、それらの行での文字切り出しを有効にした。

文字を切り出せなかった割合が4.1%と高いのは、普通に入手できる新聞紙面の品質がよくないことが最大の原因であると考えられる。紙面にシミなどの汚れが加わったり、何らかの原因で文字の線切れやかすれが生じると、標準的な1文字分の大きさをもつ8連結ブロックの数が1行中に少なくなり、文字切り出しの手掛りが少なくなる。これを避けるためには、紙面の品質のよい新聞を用意するのが最良の方策である。しかし、そのようなことは一般に期待できないので、何らかの画質改善処理を紙面に加える必要があろう。孤立点の除去処理はすでに行っているが、それ以上に有効な画質改善策については、現在検討中である。

“二”、“三”のように縦方向に分離した数画から成る文字を含む行では、縦方向の長さが短い8連結ブロックが続く。このような場合、追い込みの対象となった文字がその付近にあることがわかっていても、もっともらしい文字の切り出し方がいくつかあることになるので、いずれが正しいと決定することは不可能である。文字認識機構を準備して後戻り処理を行えば解決できる場合もあるが、一般にはより簡単な機構を用意

して解決すべき問題であろう。

著者らの手法は、いくつかの検討すべき問題を残したとはいえ、ほぼ満足の得られるものになっているといえよう。

謝辞 本研究の推進に当たり、貴重な助言をいただいた本講座大学院生上原邦昭氏および河合和久氏に感謝します。また討論いただいた情報処理機械学講座の皆さまに深く感謝します。

参 考 文 献

- 1) 豊田順一：ページ読み取りシステムにおける記事の追跡について，電気4学会関西連大(55) G8-34 (1980).
- 2) 坂井利之，稲垣耕作，加藤俊一：複雑な構造をもつ文書画像の自動解析，情報処理学会第23回全国大会 6C-2 (1981).
- 3) Toyoda, J., Noguchi, Y. and Nishimura, Y.: Study of Extracting Japanese Newspaper Article, Proc. of 6-th ICPR, München (1982).
- 4) 野口要治，豊田順一：新聞記事の切り抜きを行なうシステムに関する基礎的研究，情報処理学会第23回全国大会 6C-1 (1981).
- 5) 目黒真一，梅田三千雄：マルチフォント印刷漢字の認識，電子通信学会論文誌 (D), Vol. J 65-D, No. 8, pp. 1026-1033 (1982).
- 6) 西村 康，野口要治，豊田順一：新聞記事の本文を構成する文字の切りだし，情報処理学会第24回全国大会 3E-7 (1982).
- 7) 森 健一，坂井邦夫：日本語文書を読取る印刷漢字認識装置，大型プロジェクトパターン情報処理システム講演会論文集，pp. 33-44 (1977).
- 8) 秋山照雄，増田 功：印刷物の記事領域における文字の切出し，電子通信学会技術研究報告，PRL 80-70 (1980).
- 9) 小川秀夫，谷口慶治：弛緩法によるストローク分解と文字列のセグメンテーション，電子通信学会技術研究報告，PRL 80-5 (1980).
- 10) 浦 昭二，芳賀敏郎，昆野誠司：新聞のコンピュータ製作，*bit*, Vol. 12, No. 12, No. 14, Vol. 13, No. 7, No. 2 (1980-1981).

(昭和57年9月27日受付)

(昭和58年1月17日採録)